

Towards optimal explicit time-stepping schemes for the gyrokinetic equations

H. Doerk and F. Jenko*

Max-Planck-Institut für Plasmaphysik, Boltzmannstraße 2, D-85748 Garching, Germany

Abstract

The nonlinear gyrokinetic equations describe plasma turbulence in laboratory and astrophysical plasmas. To solve these equations, massively parallel codes have been developed and run on present-day supercomputers. This paper describes measures to improve the efficiency of such computations, thereby making them more realistic. Explicit Runge-Kutta schemes are considered to be well suited for time-stepping. Although the numerical algorithms are often highly optimized, performance can still be improved by a suitable choice of the time-stepping scheme, based on spectral analysis of the underlying operator. Here, an operator splitting technique is introduced to combine first-order Runge-Kutta-Chebyshev schemes for the collision term with fourth-order schemes for the remaining terms. In the nonlinear regime, based on the observation of eigenvalue shifts due to the (generalized) $E \times B$ advection term, an accurate and robust estimate for the nonlinear timestep is developed. The presented techniques can reduce simulation times by factors of up to three in realistic cases. This substantial speedup encourages the use of similar timestep optimized explicit schemes not only for the gyrokinetic equation, but also for other applications with comparable properties.

Keywords: Gyrokinetic simulation, plasma turbulence, optimized explicit Runge-Kutta schemes, operator splitting, eigenvalue computation, spectral analysis

1. Introduction

Gyrokinetic simulation codes are a common tool for obtaining *ab-initio* predictions of turbulence properties in strongly magnetized high-temperature plasmas.[1, 2] Such plasmas are present in magnetic confinement fusion devices, and in astrophysics. Gyrokinetic theory describes the time evolution of each species' particle distribution function f in five-dimensional phase space (one velocity space variable, the gyro-angle, is averaged out). Obtaining a solution of this nonlinear partial integro-differential equation generally requires high-performance computing. In the past decades, gyrokinetic codes have become substantially more realistic by applying higher numerical resolution and by moving to more comprehensive physics models. For example, the effect of collisions is formally weak in dilute high-temperature plasmas and thus has often been neglected. Today, one realizes that including a suitable collision operator in gyrokinetic turbulence is not only required for a physically correct entropy balance,[3] but can also greatly influence the turbulence level—through damping of zonal flows—or even change the turbulence regime by modifying the growth rate of certain types of microinstabilities.[4, 5, 6, 7, 8] Since more realistic physics models require increased computational effort, progress is enabled by the availability of more powerful computers and by the use of advanced algorithms, the importance of the latter often being underestimated.

Three classes of gyrokinetic turbulence codes (particle-in-cell, semi-Lagrangian, and Eulerian) exist. Here, the Eulerian

approach, which became popular approximately fifteen years ago, is considered. Several major code projects exist in this area, for instance GENE [9, 10, 11, 12], GS2 [13, 14], GYRO [15, 16], GWK [17], and AstroGK [18]. The common basic procedure is the so-called method of lines: After discretizing phase space on a fixed grid, the resulting large system of ordinary differential equations is evolved with a time integration scheme. However, the choice of algorithms can differ substantially. Besides various possible choices for phase space grids and the representation of derivatives on those grids, time discretization is performed in several ways, see Ref. [19] for a useful overview. Operator splitting techniques for the collisional term are used in GYRO, GS2 and AstroGK. Some codes (like GS2) even choose to split off the nonlinear term from linear dynamics, while others avoid splitting to treat these terms on an equal level. Moreover, implicit, as well as explicit schemes are applied. While GS2 (and AstroGK) treat all linear terms implicitly, the GYRO algorithm splits off fast linear terms (the parallel electron dynamics) in an implicit-explicit (IMEX) fashion. Here, we focus on fully explicit time integration, as employed in GENE and GWK, for example. Explicit methods offer the advantages of an excellent performance on massively parallel systems and the straightforward implementation of nonlinear terms. The drawback is a strict stability limit that is set on the timestep Δt , which depends on the fastest dynamics in the system. A major advance from gyrokinetic theory is to analytically remove extremely fast timescales like compressional Alfvén waves or particle gyromotion, leaving only relevant dynamics and enabling an explicit treatment. One of the fastest remaining terms is then given by the (generalized) nonlinear drift velocity

*corresponding author

Email address: Hauke.Doerk@ipp.mpg.de (H. Doerk and F. Jenko)

$v_\chi = E_\chi \times B$ that combines electric and magnetic field fluctuations. When this nonlinear advection limits the timestep according to a Courant-Friedrichs-Lewy (CFL) relation $\Delta t \lesssim \Delta x/v_\chi$, [20] fully explicit schemes are likely to be the more efficient choice (particularly in view of increasing problem size).[9]

It is sometimes stated that collisions require an implicit treatment, since the explicit diffusive timestep limit would be too strict.[21] However, we find severe restrictions only for rather large collision frequencies (in the tokamak edge, for example) or for very high velocity resolution. In this work, we introduce a splitting scheme involving Runge-Kutta-Chebyshev (RKC) schemes with extended real stability boundary,[22, 23] which enables an explicit treatment of a sophisticated collision operator even in these extreme cases. Partitioned RKC schemes have recently been developed which are also stable for advective terms, involving, however, a larger number of operator evaluations per step.[24, 25]

In principle, accuracy limits can also be imposed on the timestep. In this context, we note that the overall numerical accuracy of gyrokinetic simulations is generally strongly restricted by the grid resolution in five-dimensional phase space. A relative error tolerance of approximately 10^{-3} is already considered to be sufficient, even for linear simulations. Nonlinear simulations are subject to statistical errors of the order of 10%, underlining the fact that long simulation times rather than highly accurate steps are needed. In consequence, the use of low-order time integration schemes is well justified to speed up computations.

In this paper, a detailed analysis of the spectral properties of the discretized system allows us to identify a class of highly efficient first-order explicit schemes (with largely extended stability boundaries), which we apply to the gyrokinetic code GENE. The remainder of this paper is organized as follows. The relevant equations are summarized in Section 2 and timestep limiting physics is discussed. In Section 3 we introduce relevant explicit RK schemes and review their stability conditions. In Section 4, the efficiency and accuracy of splitting techniques are discussed, which allow time-stepping schemes to be tailored to the individual parts of the operator. Finally, in Section 5 we address the timestep restrictions in nonlinear simulations. We show that the $E_\chi \times B$ advection shifts the eigenvalues along the imaginary axis, which is relevant for the stability limit. This observation forms the basis of a greatly improved estimate of the nonlinear timestep. Overall, these two methods of (i) operator splitting and (ii) an improved timestep estimate enhance the code efficiency by up to a factor of three in realistic cases. Since the code was already highly optimized, this speedup is significant.

2. The gyrokinetic equations

The gyrokinetic equation

$$\partial_t g = G[t, g] = N[\bar{\chi}, g] + L[g] + C[g] \quad (1)$$

describes the time evolution of the (modified) perturbed gyrocenter distribution g for each plasma species in $\{x, y, z, v_\parallel, \mu\}$

phase space. The notation

$$\bar{\chi} = \bar{\phi}_1 - v_\parallel \bar{A}_{1\parallel} + \mu \bar{B}_{1\parallel} \quad f = g + v_\parallel \bar{A}_{1\parallel} F_0$$

$$F_0 = \frac{n_0}{\pi^{3/2} v_T^3} \exp \left[\left(v_\parallel^2 + \mu B_0 \right) / T_0 \right].$$

introduces the fluctuating potential $\bar{\chi}$, consisting of electrostatic perturbations $\bar{\phi}_1$ and magnetic perturbations $\bar{A}_{1\parallel}$ and $\bar{B}_{1\parallel}$, where the overbar denotes a gyroaverage. The gyrocenter distribution is split into a background (Maxwellian) distribution F_0 and a small fluctuating part f . The background magnetic field is B_0 and the background density n_0 , temperature T_0 , thermal velocity $v_T = (2T_0/m)^{1/2}$ and particle mass m are given for each plasma species. The gyrokinetic version of Maxwell's equations is used to compute a self-consistent fluctuating potential from g , which closes the system of equations. We refer to Refs. [2, 10, 12] for a detailed description and derivation.

Eq. (1) is symbolically written as the sum of three integro-differential operators whose physical meaning is briefly discussed in the following. The linear terms $L[g]$ contain parallel advection along the magnetic field lines, as well as perpendicular drifts such as curvature and ∇B drifts, and temperature and density gradient terms. The nonlinear term $N[\bar{\chi}, g]$ describes turbulent re-distribution of free energy due to perpendicular $E_\chi \times B$ advection, where the generalized fluctuating field is defined as $E_\chi = -\nabla \bar{\chi}$. Finally, the linearized Landau-Boltzmann collision operator $C[g]$ describes diffusion and dynamical friction in velocity space, including back-reaction terms that ensure conservation of particles, momentum, and energy. Details of the implementation of the collision operator in GENE can be found in Refs. [26, 27].

For numerical solution, Eq. (1) is discretized on a fixed grid in phase space, where common techniques from computational fluid dynamics, such as spectral methods, finite differencing, finite element, and finite volume schemes can be used. This results in a large system of ordinary differential equations for the time evolution of the state vector g . When non-dissipative differencing schemes are employed, as is the case with the GENE code, it may be necessary to add hyperdiffusion terms to $L[g]$ that remove unphysical grid-size oscillations in some phase space directions.[28, 29]

One way of solving this space-discretized system is to perform initial value computations, for which we consider Runge-Kutta (RK) schemes here. In the nonlinear case, we desire to find a statistically stationary turbulent state. Linear initial value computations yield the fastest growing solution (sometimes referred to as a mode), which constitute the driving force for plasma turbulence and are thus of great interest. Typical growth rates and frequencies are of the order of c_s/L_{ref} , where $c_s = (T_e/m_i)^{1/2}$ denotes the ion sound speed and L_{ref} is a typical macroscopic scale length, often set to the tokamak major radius. Additionally, the linearized system can be formulated as an eigenvalue problem. In this context, GENE features the use of optimized iterative algorithms provided by the SLEPc package,[30, 31, 32, 33, 34] which select a subset of eigenvector-eigenvalue pairs $\{g_i, \lambda_i\}$ that fulfill some user-specified criteria. For convenience, we split the complex eigenvalue $\lambda = \gamma + i\omega$ into a growth rate γ and a frequency ω . The

eigenvalues of largest magnitude $|\lambda_i|$ are quickly found (for example by Krylov-Schur subspace iteration), which proves extremely useful for the exact computation of the maximum stable timestep for initial value simulations. Due to the shape of the spectrum, obtaining the fastest growing solution with SLEPC is more cumbersome, but can still be faster than a corresponding initial value simulation. Moreover, subdominant and marginally stable solutions only become accessible by such eigenvalue computations. Finally, GENE can also compute the full spectrum (using ScaLAPACK routines), but this is only feasible for small problems.

As we will see in Sec. 3, the maximum stable timestep for Runge-Kutta methods is determined by the spectral properties of the underlying operator. Focussing on the linear case first, either the fastest oscillating ω_{\max} or the most damped $\gamma_{\min} = \min[\text{Re}(\lambda)]$ solutions are typically most restrictive. Let us briefly summarize physical mechanisms behind these extreme eigenvalues. Importantly, the integro-differential character of the parallel advection term $\partial_t g = v \partial_z g + v \partial_z \chi$ does not allow for a rigorous CFL approach of the form $\lambda_{\max} = k_{\max} v$, since χ is computed from g integrals. Here, v is an advection velocity and k_{\max} is the largest wavenumber in the system. A popular example for the origin of very high-frequency (and timestep limiting) solutions are kinetic shear Alfvén waves. In simplified slab geometry (and in the relevant low- β_e limit), the dispersion relation reads

$$\omega_{\text{KSA}}^2 = \frac{1}{\mu_e (k_{\perp} \rho_s)^2 + \beta_e} k_{\parallel}^2 c_s^2, \quad (2)$$

where k_{\perp} is a perpendicular wavenumber and k_{\parallel} is a parallel wavenumber.[35, 36] Here, the electron to ion mass ratio $\mu_e = m_e/m_i$, the electron beta $\beta_e = 4\pi n_e T_{e0}/B_0^2$, the ion sound gyroradius $\rho_s = c_s/\Omega_i$ and the ion cyclotron frequency $\Omega_i = (eB_0)/(m_i c)$ are introduced. The β_e parameter controls the response in Ampère's law, whereas electrostatic models use $\beta_e = 0$. We observe that as β_e approaches zero, the frequency $\omega_{\text{KSA}} \sim k_{\parallel}/(k_{\perp} \rho_s)$ can become very large. Indeed, setting $k_{\parallel \max} L_{\text{ref}} \sim \pi/\Delta z \sim 10$, $k_{\perp} \rho_s \sim 0.05$, $\mu_e = 1/3600$ and $\beta_e = 0$ we obtain $\omega_{\text{KSA}} \sim 4000 (c_s/L_{\text{ref}})$, about four orders of magnitude larger than the typical values for growth or damping rates. Fortunately, even small values of β_e prevent the divergence of ω_{KSA} , so that it can be beneficial to include electromagnetic effects for kinetic electron simulations, even if the dominant physics is of electrostatic nature. In the opposite limit of $\beta_e/\mu_e \gtrsim (k_{\perp} \rho_s)^2$, which is more relevant to actual fusion plasmas, Eq. (2) transitions into the classical Alfvén wave dispersion relation $\omega_A^2 = v_A^2 k_{\parallel}^2$ with $v_A = c_s/\beta_e^{1/2}$ denoting the Alfvén velocity. Also the parallel streaming of electrons is often relevant, even if field-aligned coordinates are used. The characteristic frequency is given as $\omega_{\parallel} \sim k_{\parallel} v_{\parallel e}$, which can be linked to a CFL condition. In typical fusion experiments, the electron thermal velocity is larger or comparable to the Alfvén velocity ($v_{\parallel e}/v_A \sim v_{te}/v_A \sim (\beta_e/\mu_e)^{1/2} \gtrsim 1$), so that the CFL condition for kinetic electrons is usually more restrictive than the limit due to Alfvén waves.

A third notable source of high frequency solutions is linked to magnetic curvature and ∇B drifts, which are (roughly) pro-

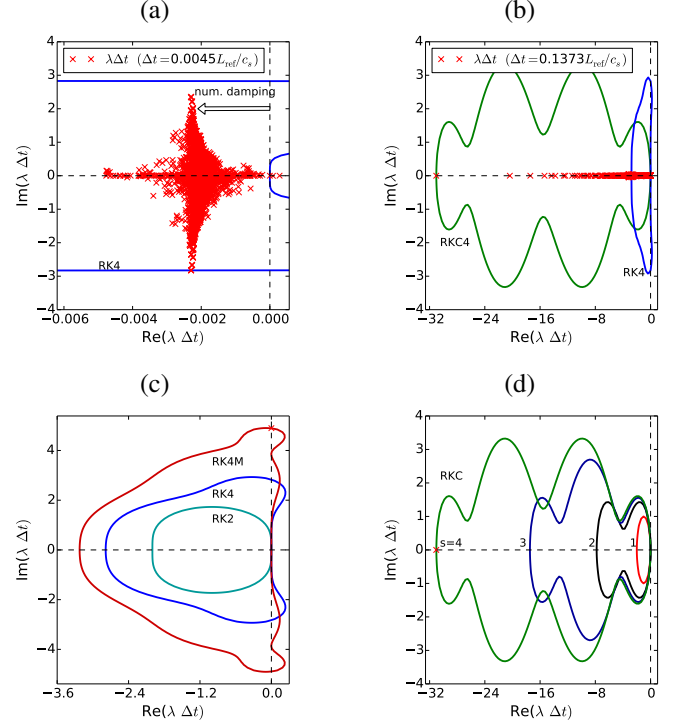


Figure 1: (a) Spectrum of the collisionless, linearized gyrokinetic operator L , scaled with Δt_{RK4} and (parts of) the stability boundary of the RK4 scheme. Note that $\text{Im}(\lambda) \gg \text{Re}(\lambda)$. (b) Spectrum of the collision operator, scaled with Δt_{RKC4} and stability boundaries of the RK4 and RKC4 schemes (for $\Delta t = \Delta t_{\text{RKC4}}$). Here, the eigenvalues have a dominating real part, where the RKC4 stability region has a much larger extent. (c) Stability boundaries of the RK2, the RK4 and the RK4M schemes. (d) First-order RKC stability boundaries up to four stages.

portional to particle energy $\epsilon = mv^2/2$ and perpendicular wavenumber k_{\perp} . Thus, if either highly energetic particles or very high wavenumbers are involved, these drifts are expected to play a relevant role.

Of course, the above considerations are based on simplified versions of the gyrokinetic equation, or even on single terms. In the general, more comprehensive case, all these terms are coupled and one has to numerically compute the spectrum. The result (using ScaLAPACK) is shown in Fig. 1(a) for the discretized, collisionless linear gyrokinetic operator. Indeed, the eigenvalue spectrum is stretched along the imaginary axis. One generally finds the highest frequency either at smallest or largest wavenumber k_y , which is consistent with the phenomena discussed above. In the appropriate limits, the magnitude of the numerically computed frequency scales as expected. While hyperdiffusion terms can be necessary to stabilize spurious grid-size oscillations, their effect on the eigenvalue spectrum is just a comparably small shift along the negative real axis, which has no big impact on stability considerations. It is important to note that upwind (or other dissipative) discretization methods can strongly distort the spectrum, and that dissipation on low k_y potential fluctuations ϕ can cause large negative eigenvalues and should thus be avoided.[36]

In the flux-tube limit, collisional diffusion (a sink of fluctu-

ation entropy) provides the only physically motivated damping in the gyrokinetic system.[37] The numerically computed spectrum of the collision operator is shown in Fig. 1(b); the eigenvalues are distributed along the negative real axis.¹ The maximum damping rate is proportional to the collision frequency ν_c and roughly proportional to $\Delta v_{\parallel}^{-2}$, which reflects the diffusive character of this term. Since the velocity-dependent collision rate diverges for $v = 0$, it is essential to distribute v_{\parallel} grid points symmetrically around $v_{\parallel} = 0$, with no grid point at the origin. Even if this is done, collisions can dominate the timestep, particularly towards the edge of tokamak devices (like ASDEX Upgrade [38]), where the collisionality is larger compared to core plasmas, and stronger flux-surface shaping can require a finer velocity grid.

Finally, including the nonlinear $E_{\chi} \times B$ term

$$\begin{aligned} N[\bar{\chi}, g] &= \frac{c}{C_{xy}} \frac{B_0}{B_{0\parallel}^*} (\partial_y \bar{\chi} \partial_x g - \partial_x \bar{\chi} \partial_y g) \\ &\equiv v_{\chi}^x \partial_x g + v_{\chi}^y \partial_y g \end{aligned} \quad (3)$$

prohibits a direct eigenvalue computation, since the advection velocities v_{χ}^x and v_{χ}^y are computed self-consistently from g . In such cases, a common technique is to freeze v_{χ} at time t_n (and maximize it in space) for computing a timestep estimate. Details on this procedure and the combination of nonlinear and linear stability limits are given in Section 5. Due to the advective character of $N[\bar{\chi}, g]$, a frequency shift along the imaginary axis is found.

3. Stability properties of relevant explicit Runge-Kutta methods

According to the method of lines the nonlinear, time-dependent operator $G[t, g]$ is discretized on a fixed grid in phase space, which turns Eq. (1) into a large set of first-order ordinary differential equations for the time evolution of the state vector g . In this section, explicit RK methods are considered to advance g_n at time t_n to g_{n+1} at time t_{n+1} with the timestep $\Delta t = t_{n+1} - t_n$. We focus on explicit RK schemes of the diagonal form

$$\begin{aligned} g_{n+1} &= g_n + \Delta t \sum_{j=1}^s b_j k_j \\ k_j &= G \left[t_n + a_j \Delta t, g_n + \Delta t \sum_{j'=1}^s a_{jj'} k_{j'} \right] \end{aligned} \quad (4)$$

where s is the number of stages and the coefficients fulfill $\sum_{j=1}^s b_j = 1$ as well as $a_1 = 0$. In this simplified scheme, only k_{j-1} is used for computing k_j , while in general, all $k_{j'}$ with $j' < j$ can be allowed to contribute. Obviously, this procedure is memory efficient, since only up to three additional vectors of the size of g have to be stored. The order of consistency p is

¹Imaginary parts that arise from discretization and boundary conditions are negligible.

determined by comparing Eq. (4) with a Taylor expansion

$$\begin{aligned} g_{n+1} &= g_n + \sum_{j=1}^s c_j \Delta t^j \frac{d^j}{dt^j} g \Big|_{t_n} \\ &= g_n + \sum_{j=1}^p \frac{\Delta t^j}{j!} \frac{d^j}{dt^j} g \Big|_{t_n} + \sum_{j=p+1}^s c_j \Delta t^j \frac{d^j}{dt^j} g \Big|_{t_n} \end{aligned} \quad (5)$$

of g about t_n , where the $\{c_i\}$ can be computed from the coefficients $\{a_j, b_j\}$ in Eq. (4). The required order of consistency $p \leq s$ thus imposes constraints on the $\{a_j, b_j\}$. For the linear problem, one inserts the eigenvalue equation $dg/dt = \lambda g$ into the Eq. (5) to obtain the stability polynomial

$$P_s(\lambda \Delta t) = 1 + \sum_{j=1}^s c_j (\lambda \Delta t)^j,$$

allowing to write down the RK stability condition

$$|P_s(\lambda \Delta t)| \leq 1 \quad (6)$$

that must be fulfilled for all λ in the left complex half-plane ($\gamma \leq 0$) to ensure that these actually stable or damped solutions are not artificially destabilized by the explicit scheme. Thus, a sufficiently small timestep must be chosen. We define β_{imag} (β_{real}) to be the extent of the stability boundary along the imaginary (negative real) axis, i.e. $|P_s(-\beta_{\text{real}})| = |P_s(i\beta_{\text{imag}})| = 1$. For instance, in the case of a simple advection problem $\partial_t g = v \partial_x g$ the eigenvalues $\lambda = ik_x v = i\omega$ are imaginary, and the maximum timestep is $\Delta t = \beta_{\text{imag}}/\omega_{\text{max}}$. For maximum time order schemes ($p = s$), the total stability region is completely determined by the number of stages. Increasing the number of stages above p requires additional evaluations of $G[g]$, but adds free parameters for shaping the stability polynomial to lower timestep constraints and reduce the overall computational cost.

Internal stability is found to become increasingly important at a large number of stages.[22, 23] Since we use diagonal methods with no more than six stages, no restrictions are found in practice.

Among other choices, GENE features the use of standard $s = p$ second-order (RK2) and fourth-order (RK4) schemes, as well as an optimized fourth-order scheme (RK4M) with six stages, following Ref. [39]. The corresponding coefficients are given in Table 1 and the stability boundaries are depicted in Fig. 1(c). Additionally, a class of s -stage Runge-Kutta-Chebyshev (RKC s) schemes is considered, which are unconditionally unstable for (undamped) waves, but are powerful in the case of the real spectrum of the collision operator. The Chebyshev polynomials are defined as

$$\begin{aligned} T_0(x) &= 1 \quad T_1(x) = x \\ T_j(x) &= 2x T_{j-1}(x) - T_{j-2}(x), \quad j \geq 2. \end{aligned}$$

Restricting ourselves to first order of consistency $p = 1$, shifted Chebyshev polynomials possess the optimal stability along the negative real axis with $\beta_{\text{real}} = 2s^2$. In order to stabilize small imaginary parts of the $\{\lambda_i\}$, the damped shifted Chebyshev polynomials

$$P_{\text{RKC}1}(z) = T_s(\omega_0 + \omega_1 z)/T_s(\omega_0) \quad (7)$$

	a_1	a_2	a_3	a_4	a_5	a_6
RK2	0	0.8				
RK4	0	0.5	0.5	1.0		
RK4M	0	0.16791847	0.4829844	0.7054607	0.0929587	0.7621008
	b_1	b_2	b_3	b_4	b_5	b_6
RK2	0.375	0.625				
RK4	1/6	1/3	1/3	1/6		
RK4M	-0.15108371	0.7538468	-0.3601660	0.5269677	0.0	0.2304351

Table 1: Coefficients for a two stage RK2, a four stage RK4 and a six stage RK4M method.

	a_1	a_2	a_3	a_4
RKC1	0	1		
RKC2	0	0.1706953512		
RKC3	0	0.03569626261	0.1377742151	
RKC4	0	0.03221719644	-1.987635269	0.03221719644
	b_1	b_2	b_3	b_4
RKC1	1			
RKC2	0.2497313672	0.7502686328		
RKC3	0.1115050079	-0.2891490316	1.177644024	
RKC4	0.001610859822	0.11936272179	-0.2816076282	1.160634047

Table 2: Coefficients for the various (diagonal) first-order Runge-Kutta-Chebyshev methods implemented in the GENE code.

have been introduced in terms of coefficients $\omega_0 = 1 + \epsilon/s^2$ and $\omega_1 = T_s(\omega_0)/T'_s(\omega_0)$. [23, 22] Setting the small parameter $\epsilon > 0$ then introduces damping. Here, $\epsilon = 0.05$ is chosen, which yields a stability boundary of $\beta_{\text{real}}(s) \approx 1.93s^2$. The first-order RKC coefficients implemented in GENE are summarized in Table 2 and the stability boundaries are shown in Fig. 1(b) and (d). One recognizes the RKC1 scheme to be identical to the explicit Euler scheme. The remaining RKC schemes deviate from the ones described in Ref. [23]. While our (diagonal) approach is more memory efficient, we lose the opportunity of recursively defining internally stable schemes for an arbitrary number of stages. However, we observe in the following sections that at most four stages are necessary in our case.

4. Timestep optimization with an operator splitting technique

In the previous sections, we have introduced various explicit RK schemes and discussed properties of the three operators $L[g]$, $N[\bar{\chi}, g]$ and $C[g]$ of Eq. (1) that determine the stability of these schemes. In the following, we attempt to find efficient RK schemes for the individual operators. In Sec. 2 it has been shown that for the collisionless part ($N + L$) an extended stability β_{imag} along the imaginary axis is required. If accuracy constraints can be ignored, computational efficiency can be characterized by the ratio $\beta(s)/s$, as $\beta(s)$ sets the maximum timestep and s measures the cost per step. Interestingly, among all 4-stage methods the fourth order (RK4) scheme with $\beta_{\text{imag}} = 2\sqrt{2}$ is optimal in that respect. The (RK4M) $s = 6$ scheme has $\beta_{\text{imag}} = 4.90$ and thus is about 15% more efficient than (RK4) and about 7% more efficient than the (RK4(3)5[2R+])C scheme referred to in Ref. [15]. To our knowledge, only a theoretical

upper bound of $\beta_{\text{imag}} \leq s$ exists for (even) $s > 4$, and (RK4M) is only 18% lower than that. [40]

The eigenvalues of the collision operator $C[g]$, on the other hand, extend along the negative real axis. In this case, the RKC methods discussed in Sec. 3 possess a near-optimal stability polynomial with a computational efficiency $\beta_{\text{real}}/s \approx 1.93s$ that increases linearly in the number of stages.

Although it is possible to include the collision operator in L and perform time integration with a RK4 method, the strong benefits of RKC methods can only be exploited when an operator splitting technique is applied. In exponential notation, it can easily be shown that the symmetric (Strang) splitting

$$\begin{aligned} g_{n+1} &= e^{\Delta t C/2} e^{\Delta t (N+L)} e^{\Delta t C/2} g_n \\ &= e^{\Delta t [C+N+L]} g_n + O(\Delta t^3) \end{aligned}$$

is second-order accurate in Δt . [41] In contrast, the non-symmetric splitting

$$g_{n+1} = e^{\Delta t C} e^{\Delta t (N+L)} g_n = e^{\Delta t [C+N+L]} g_n + O(\Delta t^2)$$

is formally only first-order accurate. However, when the propagation of g with the first operator (C) does not change the second operator ($N + L$), the second half-step can be combined with the first half-step of the next time iteration. In this case, both of the above splitting schemes are of second order. [42] In gyrokinetics, this argument holds for linear computations only. In the nonlinear case, applying collisions on g does generally alter the self-consistent potentials computed with Maxwell's equations, which in turn changes the nonlinear operator N . In consequence, second-order accuracy in nonlinear simulations is expected only for the symmetric splitting. Since the RKC

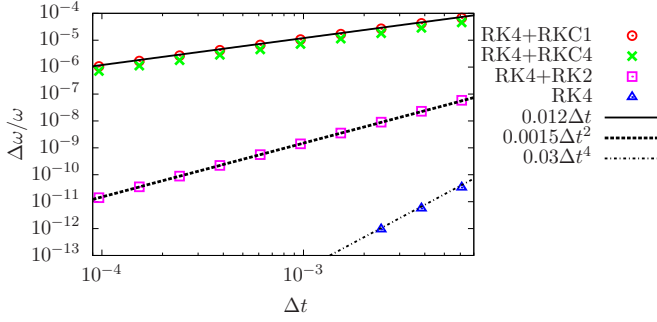


Figure 2: The frequency error $\Delta\omega/\omega$ of various explicit RK schemes is shown as a function of the timestep. The order of convergence in Δt is illustrated by straight lines in the double logarithmic plot. As expected, the error of the RK4 scheme (blue triangles) is $O(\Delta t^4)$. The error of the operator splitting method Eq. (8) is found to be $O(\Delta t^2)$, as can be observed when second or higher-order individual schemes are used (magenta squares). Choosing first-order RKC schemes (red dots $s = 1$, green crosses $s = 4$), the convergence order reduces to $O(\Delta t)$, but the small prefactor ($\approx 10^{-2}$) ensures acceptable accuracy.

schemes that we consider here are only first-order accurate anyway, we can choose the simple approach

$$\begin{aligned} g_{n+1}^{\text{vl}} &= \mathcal{RK}^{\text{vl}} \{L[g_n] + N[\bar{\chi}_n, g_n], \Delta t\} \\ g_{n+1} &= g_{n+1}^{\text{vl}} + \mathcal{RK}^c \{C[g_{n+1}^{\text{vl}}], \Delta t\} \end{aligned} \quad (8)$$

of alternating propagation with the collisionless (Vlasov) operator $(L + N)$ and the collision operator C , using a common timestep. The time-stepping schemes $\mathcal{RK}^{\text{vl}}\{\cdot, \Delta t\}$ and $\mathcal{RK}^c\{\cdot, \Delta t\}$ can now be chosen individually. Scanning the timestep for different choices with the GENE code, Fig. 2 confirms the above considerations on a simple test case of an ion temperature gradient driven (ITG) mode, with kinetic electrons and $\beta_e = 0.1\%$. The frequency error $\Delta\omega = \omega - \omega_{\text{conv}}$, compared to the converged RK4 result is measured with a precision of 5×10^{-13} in this example. For an understanding of the results it is important to note that a p th-order scheme has an $O(\Delta t^{p+1})$ error for computing g_{n+1} from g_n . Since the eigenvalue is basically determined by fitting an exponential as

$$\lambda + \Delta\lambda = \frac{g_{n+1} - g_n + O(\Delta t^{p+1})}{g_n \Delta t},$$

we expect p th-order convergence for the frequency error $\Delta\omega$. As a side note, also the global error for reaching a fixed simulation time t is $O(\Delta t^p)$, since choosing a smaller value for Δt requires an accordingly larger number of timesteps to be computed. In summary, using RKC schemes for collisions brings us back to first-order in time, as expected. However, the observed prefactor of the order of 10^{-2} in Fig. 2 is relatively small, so that an acceptable accuracy of at least 10^{-4} is obtained. Thus, in practice, no effect on the accuracy of the physically relevant solutions is visible with respect to higher-order methods, which has been confirmed for a large number of linear and nonlinear cases.

Two main advantages of applying this operator splitting are identified: (i) In cases of moderate collisionality, the one stage RKC1 scheme is sufficient for collisions, which

saves three(five) calls of $C[g]$, compared to including it in the RK4(RK4M) scheme. This remains valid in nonlinear computations, although the total fraction of CPU time spent to compute collisions will be smaller. (ii) In strongly collisional cases, the timestep is restricted by strongly damped collisional eigenvalues. Computation time can then be saved by adding RKC stages, which allows larger timesteps. In nonlinear simulations the timestep is rarely dominated by collisions, even if it is in the corresponding linear case.

Thus, it is reasonable to determine the optimal number of RKC stages dynamically, so that the maximum stable RKC timestep Δt_{RKC} is always somewhat larger than Δt_{RK4} , the maximum stable RK4 timestep. In this way, collisions never restrict the timestep. This makes sense, as long as evaluating the collisionless part dominates the computational cost, which is generally the case. For evaluating Δt_{RK4} and Δt_{RKC} , a small set of most restrictive eigenvalues of L and C , λ_L and λ_C are pre-computed with fast largest-magnitude SLEPc algorithms. When multiple Fourier modes k_y are present, we make use of the fact that L and C are block-diagonal in this dimension. In this way, only a very low percentage of the following initial value computation ($\lesssim 1\%$ of a linear run) is needed for this step.

We note, however, that the equations implemented in GENE require the computation of f (and $\bar{\chi}$) before every call of C , because these fields have to be kept consistent with g . This produces an additional overhead of the splitting scheme. In rare cases, the computation of f and $\bar{\chi}$ is found to be relatively costly, but in general this is easily over-compensated by the gain in timestep or the less frequent calls of C itself. A positive side-effect is related to the fact that in the GENE code the velocity space dimension μ is stored in the last index of g and therefore is widely spread in the system memory. Since only collisions require ghost-cells in this dimension, which are exchanged via the message passing interface library, the reduction in number of calls of C improve the parallelization efficiency.

The efficiency of the adaptive RKC operator splitting methods in combination with RK4 and RK4M is demonstrated in the following. We first focus on linear physics and compute the fastest growing solution with the initial value solver. Two typical cases that are sensitive to collisions are chosen. One is a trapped electron mode problem (TEM) in circular model geometry. The other is a microtearing mode problem (MTM) for physics parameters of the ASDEX Upgrade discharge 27963 at the radial position $\rho_{\text{tor}} = 0.85$. The eigenvalues computed by the GENE code are $\gamma_{\text{TEM}} = 0.2610 c_s/L_{\text{ref}}$, $\omega_{\text{TEM}} = -0.6380 c_s/L_{\text{ref}}$ for the TEM case, and $\gamma_{\text{MTM}} = 0.181 c_s/L_{\text{ref}}$, $\omega_{\text{MTM}} = -1.332 c_s/L_{\text{ref}}$ for the MTM case. They coincide for all schemes considered up to the given convergence accuracy of 10^{-3} . Table 3 summarizes the results for code efficiency. In both cases, operator splitting leads to a strong increase in efficiency. In the TEM case with $\nu_{ei}/\omega = 0.38$, the timestep is not limited by collisions. Here, less frequent calls of the collision operator lead to shorter runtime and thus RKC1 is most efficient. In the MTM case, the collisionality is larger ($\nu_{ei}/\omega = 1.2$), so that adding up to two or three RKC stages increases efficiency due to a gain in timestep. Interestingly, the

MTM	$\Delta t \times 100(R/c_s)$	time/s	speedup
RK4	0.107	1181	1
+RKC1	0.0772	1099	1.1
+RKC2	0.246	440	2.7
+RKC3	0.246	529	2.2
+RKC4	0.246	620	1.9
RK4M	0.124	1534	0.8
+RKC1	0.0772	1490	0.8
+RKC2	0.301	459	2.6
+RKC3	0.426	376	3.1
+RKC4	0.426	432	2.7

TEM	$\Delta t \times 100(R/c_s)$	time/s	speedup
RK4	0.149	246	1
+RKC1	0.148	152	1.6
+RKC2	0.148	207	1.2
+RKC3	0.148	266	0.9
+RKC4	0.148	309	0.8
RK4M	0.257	212	1.2
+RKC1	0.257	113	2.2
+RKC2	0.257	146	1.7
+RKC3	0.257	179	1.4
+RKC4	0.257	212	1.2

Table 3: Test cases for the use of operator splitting with the adaptive RKC method combined with the RK4 and RK4M schemes. Compared to the RK4 case without splitting, the code efficiency is enhanced by up to a factor of three, depending on the problem. Simulation times are determined on 64 CPUs of the HELIOS system.

RKC1 splitting method requires a smaller timestep (which is explained by the lower stability boundary) but still is slightly more efficient due to the reduced number of calls of the collision operator. As expected, for both cases the runtime increases as soon as the optimal number of RKC stages is exceeded.

The benefits of operator splitting with RKC schemes become even more striking when replacing the RK4 method with the optimized six-stage RK4M method that is more efficient for large imaginary eigenvalues. Again, this is attributed to the use of a larger timestep, which also results in less overall evaluations of the collision operator. In consequence, the combination of RK4M and RKC1 leads to the lowest runtime for these linear runs. Compared to the standard RK4 scheme, the speedup is a factor of two to three, depending on the parameter set. This is significant when comprehensive physics models are employed to perform large (multidimensional) parameter studies, as is routinely done in quasilinear transport predictions for fusion plasmas (see [11, 43, 44, 45], for example).

Additionally, we have compared the time traces of a nonlinear simulation of the TEM case with and without operator splitting using up to two stages. The simulation times are given in Table 4. Here, the timestep is set by linear physics of the high wavenumbers even in the nonlinear simulation. Due to reduced computational time per step, the RKC1 operator splitting method in combination with the RK4M time scheme has the largest speedup with respect to the RK4 scheme without operator splitting. We note that the physical results are identical within the statistical error bars inherent to nonlinear turbulence simulations. For the present MTM case the nonlinear terms dominate the timestep limit, as shown in the next Section, similarly to previously published MTM simulations in a slightly different parameter regime.[46] In such cases, the RKC1 collision scheme is most efficient.

5. Timestep optimization in nonlinear simulations

This section addresses the modification of the maximum stable timestep by the nonlinear term Eq. (3). For simplicity, in the present section we will include collisions in the linear operator L , unless C is separated from the RK4 scheme, as described in the previous section. In this sense, we denote Δt_L the timestep

	$\langle \Delta t \rangle \times 10^3 (R/c_s)$	CPU time/step[s]	speedup
RK4	1.74	1.08	1
+RKC1	1.74	0.74	1.45
+RKC2	1.74	0.96	1.12
RK4M	3.01	1.59	1.17
+RKC1	3.01	0.99	1.87
+RKC2	3.01	1.23	1.51

Table 4: Nonlinear simulation times for the TEM test case, measured on 512 CPUs of the HELIOS system. Since the timestep is dynamically adapted, $\langle \Delta t \rangle$ is time-averaged.

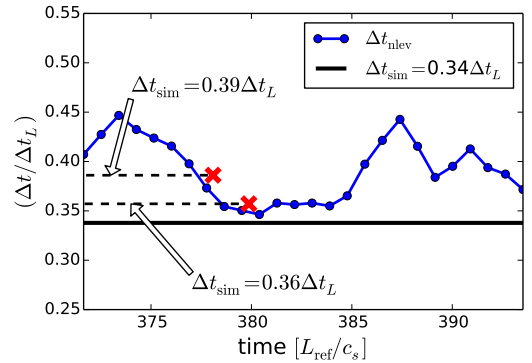


Figure 3: This figure demonstrates that the eigenvalues with linearized v_χ terms are meaningful for stability of the RK4 scheme. As an example, an electromagnetic ITG/TEM simulation in ASDEX Upgrade outer core geometry is chosen. The timesteps Δt_L and Δt_{nlev} are computed by inserting λ_L and λ_{nlev} into the stability condition Eq. (6), respectively. The linear timestep is $\Delta t_L = 0.00518 L_{ref}/c_s$ in this case. While the simulation is stable when the timestep is just below Δt_{nlev} , a numerical instability is detected in two other simulations (marked by dashed lines ending in red crosses), when Δt_{sim} exceeds Δt_{nlev} .

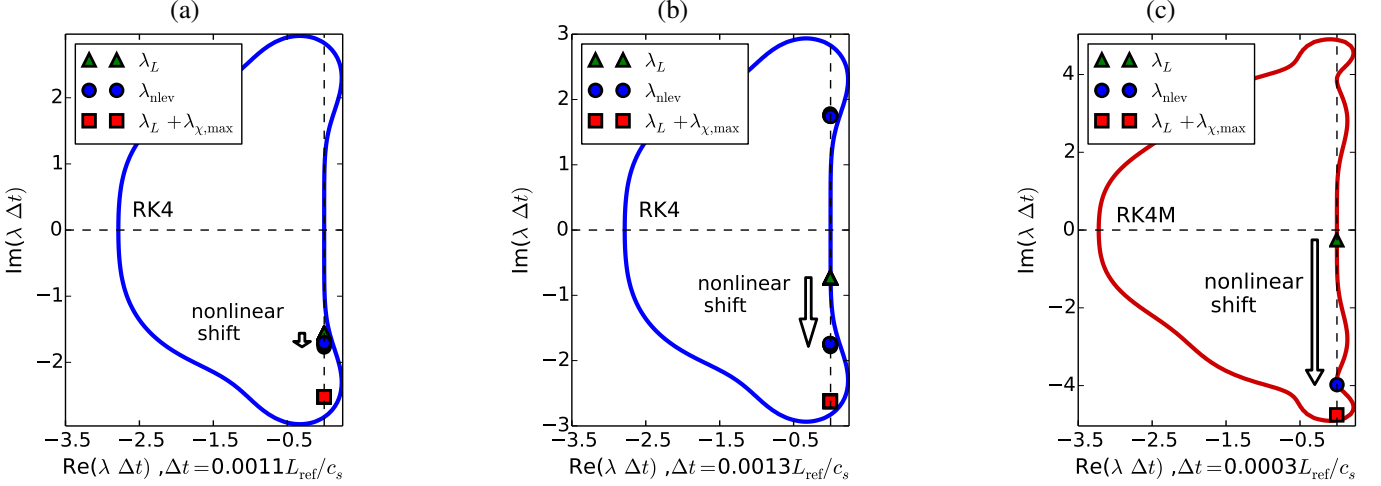


Figure 4: This figure illustrates the position of the most timestep-relevant eigenvalues λ_L and λ_{nlev} at one point in time. The sum of λ_L and the maximum nonlinear shift $\lambda_{\chi,\text{max}}$ is also depicted, accounting for the sign of λ_L . Three parameter sets with increasing nonlinear contribution are chosen. (a) is an ITG/TEM case in circular geometry, (b) and (c) use realistic tokamak geometry of the ASDEX Upgrade discharges #26459 and #27963, respectively. The shift of λ_{nlev} with respect to λ_L is along the imaginary axis, as expected. Note that Eq. (11) uses an effective eigenvalue $\lambda_{\chi,\text{max}}/c_{\text{eff}}$. Setting $c_{\text{eff}} = 0.3$ to a typical value, this lies (far) outside the figure in cases (b) and (c), which demonstrates the improvement of Eq. (10).

obtained by considering only the linear terms. The nonlinear term will be analysed in a linearized form $N[\bar{\chi}_0, g]$, which is obtained by freezing the potential to a snapshot of $\bar{\chi}_0 = \bar{\chi}(t_n)$ taken at the current time t_n . This procedure allows to access the largest magnitude eigenvalues λ_{nlev} of the combined operator $L[g] + N[\bar{\chi}_0, g]$, from which the maximum timestep Δt_{nlev} can be computed exactly from Eq. (6). Since the drift velocities v_{χ}^x and v_{χ}^y do not change much during one step, Δt_{nlev} is expected to accurately describe the stability limit of the RK scheme in the nonlinear regime. We test our hypothesis by performing three simulations with fixed timestep and identical initial condition. As illustrated in Fig. 3, the scheme becomes unstable, when Δt_{nlev} sinks below the simulation timestep. Up to this point the time-traces of physical quantities are identical, reflecting the fact that the results are converged with respect to the timestep.

Thus, it seems that we have a powerful tool at hand to exactly compute the maximum stable timestep, even in nonlinear simulations. However, performing this kind of “nonlinear eigenvalue computation” is not feasible at every timestep. In the following, we develop a fast method for approximating the influence of the nonlinear term on the eigenvalue spectrum. The nonlinearity itself constitutes a pure advection problem. We thus expect the eigenvalues to spread along the imaginary axis. Also the timestep restrictions for the linear operator stem from eigenvalues with a dominating imaginary part. Simply adding the eigenvalues is tempting, but of course the eigenvalues of the sum of two operators can not be obtained as the sum of the eigenvalues of the two separate operators, unless they are diagonalized by the same unitary transform.

Nevertheless, for combining linear and nonlinear effects, we replace Eq. (3) by

$$\begin{aligned} N[\bar{\chi}, g] &\rightarrow N_{\text{max}}[g] = i\lambda_{\chi,\text{max}}g, \\ \lambda_{\chi,\text{max}} &= |v_{\chi,\text{max}}^x|k_{x,\text{max}} + |v_{\chi,\text{max}}^y|k_{y,\text{max}} \end{aligned} \quad (9)$$

where the real number $\lambda_{\chi,\text{max}}$ maximizes the advection veloc-

ity over $\{x, y, z, \mu\}$ phase space as well as species and uses the largest wave-vectors k_x and k_y present in the simulation. Fortunately, $\lambda_{\chi,\text{max}}$ can be computed at every timestep with negligible effort. Because we are now dealing with a constant advection, the resulting total equation $\partial_t g = L[g] + N_{\text{max}}[g]$ indeed is solved by the Ansatz $g(t) = g_0 \exp[i\lambda t]$ with $\lambda = \lambda_L + \lambda_{\chi,\text{max}}$. The combined timestep $\tilde{\Delta t}_{\text{comb}}$ is defined by finding the roots of $|P_s[(\lambda_{L,\text{max}} + \lambda_{\chi,\text{max}})\tilde{\Delta t}_{\text{comb}}]| = 1$ with $\lambda_{L,\text{max}}$ being a set of most restrictive eigenvalues of L that are pre-computed before the time-stepping starts. Defining $\Delta t_{\chi,\text{max}} = \beta_{\text{imag}}/\lambda_{\chi,\text{max}}$, the combined timestep is also well captured by setting

$$\Delta t_{\text{comb}} = 1/(\Delta t_L^{-1} + \Delta t_{\chi,\text{max}}^{-1}). \quad (10)$$

For completeness we note that, in the GENE code, too frequent timestep changes are avoided by using a threshold of about 5 per cent. For Δt to be adapted, the current estimate must deviate from the present value by more than this threshold. Additionally, successive timestep increments are only allowed after a minimum of 200 steps.

The presented scheme is indeed robust under the following three conditions. (i) λ_L is dominantly imaginary. (ii) The shift of λ_L is along the imaginary axis. (iii) $\lambda_{\chi,\text{max}}$ overestimates the nonlinear shift by more than the above mentioned threshold value. The first condition (i) is justified by the use of non-dissipative differencing (plus a negligible real part caused by hyperdiffusion). On the second condition (ii) we note that the time-average of N can be modelled as a diffusive term that balances linear growth.[47] This would intuitively imply an eigenvalue shift along the negative real axis. Indeed, nonlinear $E_{\chi} \times B$ advection mediates dissipation by transporting fluctuation energy from driven phase space regions into dissipative regions. While this mechanism is important for the formation of a statistically stationary turbulent state, N is purely advective at each point in time, which is relevant for the stability of time integration. The third condition (iii) holds, because $\lambda_{\chi,\text{max}}$ is a global

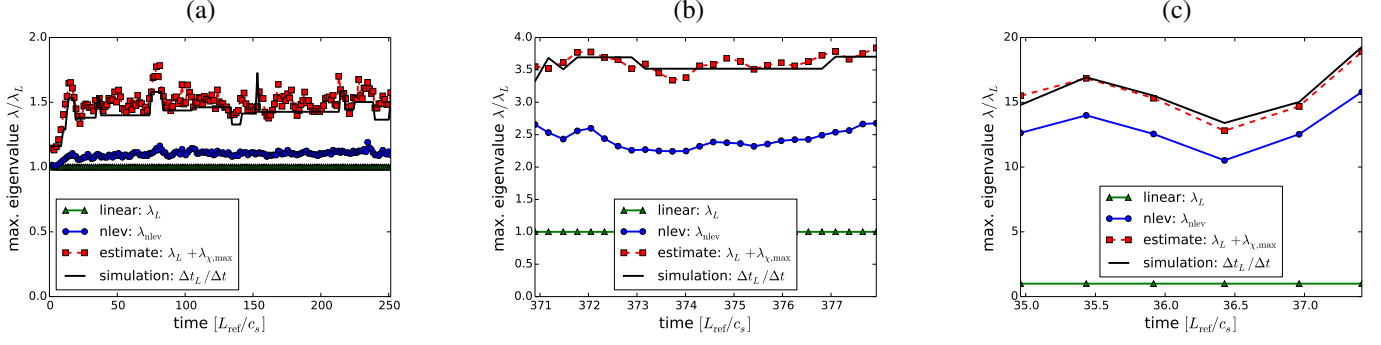


Figure 5: Time traces of linear, estimated nonlinear, and exact nonlinear eigenvalues for the three sets of parameters of Fig. 4. The sum $\lambda_L + \lambda_{\chi,\max}$ overestimates λ_{nlev} at any time, leading to a robust timestep limit. The corresponding time trace of $1/\Delta t$ does not exactly match the estimate due to built-in thresholds for timestep changes, but it stays quite close. Here, the λ_{nlev} are only computed each 1000th timestep.

maximum over phase space and thus overestimates the actual stability restriction of the nonlinear term. Also the interplay with linear terms has been observed to lower the stability constraint in some cases. However, the theoretical maximum can closely be reached (cases with dominant A_{\parallel} fluctuations and fine k_x resolution show this behavior). Nevertheless, we allow $\Delta t_{\chi,\max}$ to be multiplied with a user specified constant c_{cfl} , which can be set larger than unity in most cases. Robustness for the general case is obtained for $c_{\text{cfl}} = 0.95$ to compensate the threshold mentioned above. It should be noted that a non-spectral treatment of the x (y) dimension demands a correction factor to k_x (k_y) in Eq. (9), which can be deduced from the corresponding differencing scheme.

The shift of the eigenvalues due to the nonlinearity as well as our model are illustrated in Fig. 4 for three distinct cases with increasing nonlinear contribution. The linear eigenvalues are shifted along the imaginary axis, as expected from the advective character of N . Using the same three parameter sets, we show in Fig. 5 that the sum of the maximum linear eigenvalue and the maximum nonlinear shift $\lambda_L + \lambda_{\chi,\max}$ is indeed always overestimating the exact result λ_{nlev} . In the third case, the exact result is only overestimated by about 20% and also other cases showed almost no difference between modeled and real timestep limits. As mentioned above, other cases allow a c_{cfl} prefactor to $\Delta t_{\chi,\max}$ to be set larger than one. While this increases the efficiency, general robustness is lost. A less strict, yet robust automatic adaptation scheme would of course be desirable, but this seems to be impossible to find without performing actual expensive eigenvalue computations.

However, we want to point out that the presented estimate of Eq. (10) is already superior to the previously used CFL method

$$\Delta t_{\text{cfl}} = \min(\Delta t_L, c_{\text{cfl}} \times \Delta t_{\chi,\max}). \quad (11)$$

that uses the timestep $\Delta t_{\chi,\max}$ multiplied with a CFL constant $c_{\text{cfl}} \sim 0.3 - 0.5$ and never exceeds the linear timestep. Importantly, Eq.(11) requires c_{cfl} to be set smaller than one in order to stabilize cases of weak (but non-negligible) nonlinear influence. In the opposite limit of dominating nonlinear dynamics, Eq. (11) produces unnecessarily small timesteps. Combining the maximum eigenvalues according to Eq. (10), naturally captures both limits in a satisfactory way.

Coming back to our exemplary cases of Figs. 4 and 5, we observe that in case (a), which is an ITG/TEM case in circular geometry, the nonlinear term has only a weak influence and the improved estimate yields about the same timestep as Eq. (11) with $c_{\text{cfl}} = 0.3$ would provide. However, the latter becomes unstable for larger c_{cfl} factors, suggesting to set $c_{\text{cfl}} = 0.3$ by default. In the second example (b), which is a realistic ITG/TEM setup, Δt_{comb} is about a factor or two larger than Δt_{cfl} (with $c_{\text{cfl}} = 0.3$). Example (c) is a realistic ITG/MTM mixed case, in which the nonlinear timestep limit is even stronger, so that the linear terms play almost no role. In such cases, any $c_{\text{cfl}} < 1$ will reduce the code efficiency by about the same factor. In future high-temperature devices like ITER, the normalized fluctuation amplitude (v_{χ}) is expected to be smaller. However, the impact on the timestep is also determined by resolution settings. Beyond the examples given here, many other realistic gyrokinetic simulations show a significant nonlinear timestep restriction. Thus, our improved estimate can save a substantial amount (up to two thirds) of computation time.

6. Conclusions

In summary, we presented two methods for increasing the efficiency of gyrokinetic simulations and applied these to the plasma turbulence code GENE. First, we matched individual explicit Runge-Kutta schemes to the properties of individual parts of the equation by applying an operator splitting technique. For the collisionless part we chose classical and advanced fourth-order schemes. For collisions, we restricted ourselves to first-order Runge-Kutta-Chebyshev schemes, since they possess optimal stability properties, while higher-order schemes offer much smaller efficiency gains. Thereby, we reached an increased timestep and/or fewer evaluations of the collision operator, resulting in a speedup by a factor of up to three both in strongly and weakly collisional cases. A possible application is given by extremely large (multidimensional) parameter studies, which are, for example, needed in quasilinear transport modeling of tokamak plasmas. Time savings due to our method are striking especially in the tokamak edge, where the collisionality is increased.

Second, we investigated the impact of nonlinear advection on the timestep. Based on the observation of a frequency-shift in the eigenvalue spectrum due to the $E_{\chi} \times B$ advection velocity (which is an interesting topic in itself), we developed an improved and robust timestep estimate for nonlinear simulations. Beyond the examples shown in this paper, the new adaptation scheme has been successfully applied to a large number of simulations. Avoiding unnecessarily small timesteps, a speedup of up to a factor of two to three is realized for realistic problems. This is particularly important for large simulations including comprehensive physics, experimental plasma shaping, kinetic electrons, multiple scales, and possibly also profile variations. Constituting the high-end of fusion plasma modeling, such simulations yield the most accurate description of plasma turbulence currently available, but they are expensive: one run can consume millions of CPU hours on present-day supercomputers.

Since the choice of algorithms and their implementation are already highly optimized in GENE (as in other state-of-the-art codes), this further increase of efficiency is really significant. The techniques discussed in this work can prove extremely useful, also for other simulation codes with similar numerical schemes.

Acknowledgments

The authors would like to thank J. Abiteboul, T. Dannert, T. Görler, D. R. Hatch, F. Merz, E. Sonnendrücker, and D. Told for fruitful discussions and M. Dunne for extracting code input from the ASDEX Upgrade database. This work was supported by the Nu-FuSE project which is funded through G8 Multilateral Research by Funding Nu-FuSE grant JE 520/4-1. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement No. 277870. The numerical results presented in this work were carried out using the HELIOS supercomputer system at the Computational Simulation Centre of International Fusion Energy Research Centre (IFERC-CSC), Aomori, Japan, under the Broader Approach collaboration between Euratom and Japan, implemented by Fusion for Energy and JAEA and using the resources of the RZG computing center, Garching, Germany.

References

- [1] X. Garbet, Y. Idomura, L. Villard, T. H. Watanabe, Nucl. Fusion 50 (2010) 043002.
- [2] J. A. Krommes, Annu. Rev. Fluid Mech. 44 (2012) 175–201.
- [3] H. Sugama, T.-H. Watanabe, M. Nunami, Phys. Plasmas 16 (2009) 112503.
- [4] D. R. Ernst, P. T. Bonoli, P. J. Catto, W. Dorland, C. L. Fiore, R. S. Granetz, M. Greenwald, A. E. Hubbard, M. Porkolab, M. H. Redi, J. E. Rice, K. Zhurovich, A. C.-M. Group, Phys. Plasmas 11 (2004) 2637–2648.
- [5] D. J. Applegate, C. M. Roach, J. W. Connor, S. C. Cowley, W. Dorland, R. J. Hastie, N. Joiner, Plasma Phys. Controlled Fusion 49 (2007) 1113–1128.
- [6] Y. Xiao, P. J. Catto, W. Dorland, Phys. Plasmas 14 (2007) 055910.
- [7] W. Guttenfelder, J. Candy, S. M. Kaye, W. M. Nevins, R. E. Bell, G. W. Hammett, B. P. Leblanc, H. Yuh, Phys. Plasmas 19 (2012) 022506.
- [8] D. R. Hatch, M. J. Pueschel, F. Jenko, W. M. Nevins, P. W. Terry, H. Doerk, Phys. Plasmas 20 (2013) 012307.
- [9] F. Jenko, Comput. Phys. Commun. 125 (2000) 196–209.
- [10] F. Jenko, W. Dorland, M. Kotschenreuther, B. N. Rogers, Phys. Plasmas 7 (2000) 1904–1910.
- [11] T. Dannert, F. Jenko, Phys. Plasmas 12 (2005) 072309.
- [12] T. Görler, X. Lapillonne, S. Brunner, T. Dannert, F. Jenko, F. Merz, D. Told, J. Comput. Phys. 230 (2011) 7053–7071.
- [13] M. Kotschenreuther, G. Rewoldt, W. M. Tang, Comput. Phys. Commun. 88 (1995) 128–140.
- [14] W. Dorland, F. Jenko, M. Kotschenreuther, B. N. Rogers, Phys. Rev. Lett. 85 (2000) 5579–5582.
- [15] J. Candy, R. E. Waltz, J. Comput. Phys. 186 (2003) 545–581.
- [16] J. Candy, R. Waltz, Phys. Rev. Lett. 91 (2003) 045001.
- [17] A. G. Peeters, Y. Camenen, F. J. Casson, W. A. Hornsby, A. P. Snodin, D. Srintzi, G. Szepesi, Comput. Phys. Commun. 180 (2009) 2650–2672.
- [18] R. Numata, G. G. Howes, T. Tatsuno, M. Barnes, W. Dorland, J. Comput. Phys. 229 (2010) 9347 – 9372.
- [19] S. Maeyama, A. Ishizawa, T.-H. Watanabe, N. Nakajima, S. Tsuji-Iio, H. Tsutsui, Comput. Phys. Commun. 184 (2013) 2462–2473.
- [20] R. Courant, K. Friedrichs, H. Lewy, Math. Ann. 100 (1928) 32–74.
- [21] M. Barnes, I. G. Abel, W. Dorland, D. R. Ernst, G. W. Hammett, P. Ricci, B. N. Rogers, A. A. Schekochihin, T. Tatsuno, Phys. Plasmas 16 (2009) 072107.
- [22] P. J. van Der Houwen, B. P. Sommeijer, Z. Angew. Math. Mech. 60 (1980) 479–485.
- [23] J. Verwer, Appl. Numer. Math. 22 (1996) 359 – 379. Special Issue Celebrating the Centenary of Runge-Kutta Methods.
- [24] C. Zbinden, SIAM J. Sci. Comput. 33 (2011) 1707–1725.
- [25] A. Abdulle, G. Vilmart, J. Comput. Phys. 242 (2013) 869–888.
- [26] F. Merz, Gyrokinetic Simulation of Multimode Plasma Turbulence, Ph.D. thesis, Universität Münster, 2009.
- [27] H. Doerk, Gyrokinetic Simulation of Microtearing Turbulence, Ph.D. thesis, Universität Ulm, 2013.
- [28] M. J. Pueschel, T. Dannert, F. Jenko, Comput. Phys. Commun. 181 (2010) 1428–1437.
- [29] S. Maeyama, A. Ishizawa, T.-H. Watanabe, N. Nakajima, S. Tsuji-Iio, H. Tsutsui, Comput. Phys. Commun. 184 (2013) 2462–2473.
- [30] V. Hernandez, J. E. Roman, V. Vidal, ACM Trans. Math. Softw. 31 (2005) 351–362.
- [31] V. Hernandez, J. E. Roman, A. Tomas, V. Vidal, 2013. <http://www.grycap.upv.es/slepc>.
- [32] M. Kammerer, F. Merz, F. Jenko, Phys. Plasmas 15 (2008) 052102.
- [33] J. E. Roman, M. Kammerer, F. Merz, F. Jenko, Parallel Comput. 36 (2010) 339–358. Parallel Matrix Algorithms and Applications.
- [34] F. Merz, C. Kowitz, E. Romero, J. Roman, F. Jenko, Comput. Phys. Commun. 183 (2012) 922 – 930.
- [35] W. W. Lee, J. L. V. Lewandowski, T. S. Hahm, Z. Lin, Phys. Plasmas 8 (2001) 4435–4440.
- [36] T. Dannert, F. Jenko, Comput. Phys. Commun. 163 (2004) 67–78.
- [37] H. Sugama, M. Okamoto, W. Horton, M. Wakatani, Phys. Plasmas 3 (1996) 2379–2394.
- [38] A. Kallenbach, J. Adamek, L. Aho-Mantila, S. Äkäsloppolo, C. Angioni, C. V. Atanasiu, M. Balden, K. Behler, E. Belonohy, A. Bergmann, M. Bernert, R. Bilato, V. Bobkov, J. Boom, A. Bottino, F. Braun, M. Brüdgam, A. Buhler, A. Burckhart, A. Chankin, I. G. J. Classen, G. D. Conway, D. P. Coster, P. de Marné, R. D’Inca, R. Drube, R. Dux, T. Eich, N. Endstrasser, K. Engelhardt, B. Esposito, E. Fable, H.-U. Fahrbach, L. Fattorini, R. Fischer, A. Flaws, H. Fünfgelder, J. C. Fuchs, K. Gál, M. García Muñoz, B. Geiger, M. Gemisic Adamov, L. Giannone, C. Giroud, T. Görler, S. da Graca, H. Greuner, O. Gruber, A. Gude, S. Günter, G. Haas, A. H. Hakola, D. Hangan, T. Happel, T. Hauff, B. Heinemann, A. Herrmann, N. Hicks, J. Hobirk, H. Höhnle, M. Hölzl, C. Hopf, L. Horton, M. Huart, V. Igochine, C. Ionita, A. Janzer, F. Jenko, C.-P. Käsemann, S. Kálvin, O. Kardaun, M. Kaufmann, A. Kirk, H.-J. Klingshirn, M. Kocan, G. Kocsis, H. Kollotzek, C. Konz, R. Koslowski, K. Krieger, T. Kurki-Suonio, B. Kurzan, K. Lackner, P. T. Lang, P. Lauber, M. Laux, F. Leipold, F. Leuterer, A. Lohs, N. C. Luhmann, Jr., T. Lunt, A. Lysoivan, H. Maier, C. Maggi, K. Mank, M.-E. Manso,

- M. Maraschek, P. Martin, M. Mayer, P. J. McCarthy, R. McDermott, H. Meister, L. Menchero, F. Meo, P. Merkel, R. Merkel, V. Mertens, F. Merz, A. Mlynek, F. Monaco, H. W. Müller, M. Münich, H. Murmann, G. Neu, R. Neu, B. Nold, J.-M. Noterdaeme, H. K. Park, G. Pautasso, G. Pereverzev, Y. Podoba, F. Pompon, E. Poli, K. Polochiy, S. Potzel, M. Precht, M. J. Püschel, T. Pütterich, S. K. Rathgeber, G. Raupp, M. Reich, B. Reiter, T. Ribeiro, R. Riedl, V. Rohde, J. Roth, M. Rott, F. Rytter, W. Sandmann, J. Santos, K. Sassenberg, P. Sauter, A. Scarabosio, G. Schall, K. Schmid, P. A. Schneider, W. Schneider, G. Schramm, R. Schrittwieser, J. Schweinzer, B. Scott, M. Semp, F. Serra, M. Sertoli, M. Siccino, A. Sigalov, A. Silva, A. C. C. Sips, F. Sommer, A. Stäbler, J. Stober, B. Streibl, E. Strumberger, K. Sugiyama, W. Suttrop, T. Szepesi, G. Tardini, C. Tichmann, D. Told, W. Treutler, L. Urso, P. Varela, J. Vincente, N. Vianello, T. Vierle, E. Viezzer, C. Vorpahl, D. Wagner, A. Weller, R. Wenninger, B. Wieland, C. Wigger, M. Willensdorfer, M. Wischmeier, E. Wolfrum, E. Würsching, D. Yadikin, Q. Yu, I. Zammuto, D. Zasche, T. Zehetbauer, Y. Zhang, M. Zilker, H. Zohm, Nucl. Fusion 51 (2011) 094012.
- [39] J. L. Mead, R. A. Renaut, J. Comput. Phys. 152 (1999) 404–419.
- [40] P. Van Der Houwen, Construction of Integrational Formulas for Initial Value Problems, 1 ed., North Holland Publishing Company, Amsterdam - New York - Oxford, 1977.
- [41] G. Strang, SIAM J Numer Anal 5 (1968) pp. 506–517.
- [42] C. Z. Cheng, G. Knorr, J. Comput. Phys. 22 (1976) 330–351.
- [43] C. Bourdelle, X. Garbet, F. Imbeaux, A. Casati, N. Dubuit, R. Guirlet, T. Parisot, Phys. Plasmas 14 (2007) 112501.
- [44] F. Merz, F. Jenko, Nucl. Fusion 50 (2010) 054005.
- [45] F. Casson, R. McDermott, C. Angioni, Y. Camenen, R. Dux, E. Fable, R. Fischer, B. Geiger, P. Manas, L. Menchero, G. Tardini, the ASDEX Upgrade Team, Nucl. Fusion 53 (2013) 063026.
- [46] H. Doerk, F. Jenko, M. J. Pueschel, D. R. Hatch, Phys. Rev. Lett. 106 (2011) 155003.
- [47] F. Merz, F. Jenko, Phys. Rev. Lett. 100 (2008) 035005.